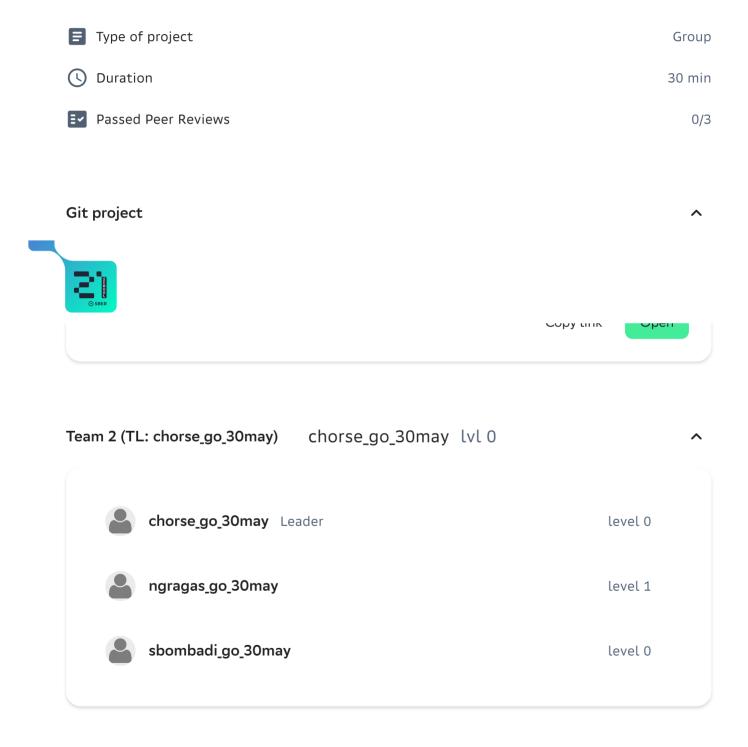
# ← Project review - Go intensive. Team01



**About** 

#### Introduction

The methodology of School 21 makes sense only if peer-to-peer reviews are done seriously. Please read all guidelines carefully before starting the review.

- Please, stay courteous, polite, respectful and constructive in all communications during t his review.
- Highlight possible malfunctions of the work done by the person and take the time to disc uss and debate it. The pedagogy is valid only if peer-evaluation is conducted seriously.
- Keep in mind that sometimes there can be differences in interpretation of the tasks and t he scope of features. Please, stay open-minded to the vision of the other.
- If you have not finished the project yet, it is compulsory to read the entire instruction bef ore starting the review.

## **Guidelines**

- Evaluate only the files that are in src folder on the GIT repository of the student or group.
- Ensure to start reviewing a group project only when the team is present in full.
- Use special flags below to report, for example, an "empty work" if repository does not contain the work of the student (or group) in the src folder of the develop branch, or "cheat" in case of cheating or if the student (or group) are unable to explain their work at any time during review as well as if one of the points below is not met. However, except for cheating cases, you are encouraged to continue reviewing the project to identify the problems that caused the situation in order to avoid them at the next review.
- Doublecheck that the GIT repository is the one corresponding to the student or the group.
- Meticulously check that nothing malicious has been used to mislead you.
- In controversial cases, remember that the checklist determines only the general order of the check. The final decision on project evaluation remains with the reviewer.

MAIN PART

# Task 00 - Scalability

Check that two programs are present, which are client and database instance

Check that you can configure replication factor when running the instance

Check that new instance fails to start if its replication factor differs from the rest of the clu ster

Check that new instance with proper replication factor successfully joins the cluster if point to it

Check that the client supports specifying hostname and port of a database instance when s tarted

Check that the client keeps track if the current instance is accessible and automatically switches over to another one if it's stopped

Check that the client is notified (and shows it to the user) in a heartbeat when current size of a cluster is (or becomes) smaller than a replication factor

# Task 01 - Balancing and Queries

Check that key is not accepted if it isn't a proper UUID4

Check that you get a proper error in a client when at some point of time no available nodes are present

Check that you can create new entries in a database without errors

Check that you can read the data from the cluster immediately after creating a new entry Check that you can read the updated data from the cluster in a couple of seconds after upd ating the existing entry

Check that you can get "Not found" response from the cluster in a couple of seconds after deleting the existing entry

Check that data is actually written to nodes in a number of copies that corresponds with a replication factor

Check that you won't get an error in a client if a current node is killed during writing, so it will just write this entry to some other node instead





### Task 02 - Long Live the King

Check that the client only interacts with a Leader node when connected to it

Check that all nodes are sending each other heartbeats with a full list of nodes

Check that new Leader is elected when current one is killed

Check that the client is able to successfully reconnect to a new Leader after election if a current one is killed

Check that replicas are now calculated on the Leader, not on the client

Check that default timeout for election is 10 seconds and can be configured

No



# **BONUS PART**

#### ^

## Task 03 - Consensus

Check that replicas are re-synced after the node which stored them dies

Check that deletion always works and there is no way a "ghost" entry can remain

Check that split-brain is not possible (the situation when election results are ambiguous an d more than one Leader are present in a cluster)

No



Fails (i)

Forbidden functions Cheat Code style Empty work

Invalid compilation Crash Leaks

Comment

Leave a comment...

✓ Review